Claims

1. A virtual circuit system comprising a circuit client for requesting establishment of a virtual circuit without human intervention from a virtual circuit manager to exchange delay sensitive data between a source node and a destination node within a packet network including a plurality of intermediate nodes selectively interconnected via a plurality of links, said circuit client comprising:

an interceptor for detecting signalling protocol data transmitted and received by said source node when said source node seeks to communicate with said destination node to facilitate a transfer of said delay sensitive data; and

a processor for deriving virtual circuit attribute data from said signalling protocol data and automatically requesting without human intervention establishment of said virtual circuit based on said virtual circuit attribute data, said virtual circuit attribute data including QoS data characterized by a specified bandwidth requirement and a specified packet transfer delay limit between said source node and said destination node.

2. The virtual circuit system of claim 1, wherein said circuit client is located within a server within said packet network and wherein said interceptor is operable to detect said signalling protocol data when said source node indirectly communicates with said destination node through said server.

3. The virtual circuit system of claim 1, wherein said circuit client is located within a proxy within said packet network and wherein said interceptor is operable to detect said signalling protocol data when said source node indirectly communicates with said destination node through said proxy, said proxy being connected to an application server operable to provide at least a portion of said virtual circuit attribute data to said proxy.

4. The virtual circuit system of claim 1, wherein said circuit client is in communication with a multimedia manager within said packet network and wherein said interceptor is operable to receive said signalling protocol data from said multimedia manager when said source node indirectly communicates with said destination node through said multimedia manager.

5. The virtual circuit system of claim 4, wherein said multimedia manager is an IP-enabled PBX.

6. The virtual circuit system of claim 1, wherein said signalling protocol data is based on H.323 protocol.

7. The virtual circuit system of claim 6, wherein said source node and said destination node are H.323 enabled, wherein said signalling protocol data is based on H.323, and wherein said circuit client operates in combination with a mediator and a H.323 protocol stack to intercept H.323 requests from said source node and to intercept H.323 responses to said H.323 requests, to process said H.323 requests and said H.323 responses to derive said virtual circuit attribute data, and to communicate with said virtual circuit manager to establish said virtual circuit.

8. The virtual circuit system of claim 7, wherein said circuit client, said mediator, and said H.323 protocol stack are located within either said source node or said destination node.

9. The virtual circuit system of claim 7, wherein said circuit client, said mediator, and said H.323 protocol stack are located within a H.323 application server, wherein said H.323 application server is in communication with either said source node or said destination node.

10. The virtual circuit system of claim 9, wherein said mediator is operable to terminate transfer of said H.323 requests and said H.323 responses if no virtual circuit can be established between said source node and said destination node.

11. The virtual circuit system of claim 7, wherein said circuit client, said mediator, and said H.323 protocol stack are located within a H.323 server proxy, wherein said H.323 server proxy is in communication with a H.323 application server and said source node.

12. The virtual circuit system of claim 11, wherein said mediator is operable to terminate transfer of said H.323 requests and said H.323 responses if no virtual circuit can be established between said source node and said destination node.

13. The virtual circuit system of claim 1, wherein said source node and said destination node are SIP enabled, wherein said signalling protocol data is based on SIP, and wherein said

circuit client operates in combination with a mediator and a SIP protocol stack to intercept SIP requests from said source node and to intercept SIP responses to said SIP requests, to process said SIP requests and said SIP responses to derive said virtual circuit attribute data, and to communicate with said virtual circuit manager to establish said virtual circuit.

14. The virtual circuit system of claim 13, wherein said circuit client, said mediator, and said SIP protocol stack are located within either said source node or said destination node.

15. The virtual circuit system of claim 13, wherein said circuit client, said mediator, and said SIP protocol stack are located within a SIP application server, wherein said SIP application server is in communication with either said source node or said destination node.

16. The virtual circuit system of claim 15, wherein said mediator is operable to terminate transfer of said SIP requests and said SIP responses if no virtual circuit can be established between said source node and said destination node.

17. The virtual circuit system of claim 13, wherein said circuit client, said mediator, and said SIP protocol stack are located within a SIP server proxy, wherein said SIP server proxy is in communication with a SIP application server and said source node.

18. The virtual circuit system of claim 17, wherein said mediator is operable to terminate transfer of said SIP requests and said SIP responses if no virtual circuit can be established between said source node and said destination node.

19. The virtual circuit system of claim 1, wherein said virtual circuit manager further includes:

a circuit network manager operable to receive a virtual circuit request from said circuit client, said virtual circuit request representing a request for a packet flow based on said QoS data between said source node and said destination node; and

a circuit domain manager operable to communicate with said circuit network manager to identify a route through said intermediate nodes from said source node to said destination node for said packet flow, and to configure each said intermediate node along said route to enable said virtual circuit, without said circuit domain manager needing to be within said

route, said QoS data being characterized by a bandwidth, a maximum allowable delay for said packet flow between said source node and said destination node, and a probability that said packet flow between said source node and said destination node will exceed said maximum allowable delay during said virtual circuit's lifetime.

20. The virtual circuit system of claim 19, wherein said circuit domain manager identifies said route based at least in part on a network topology of said packet network and whether each said intermediate node in said route will satisfy said QoS data, and wherein said circuit domain manager is operable to determine said network topology by dynamically sensing said plurality of links between said intermediate nodes directly from said intermediate nodes.

21. The virtual circuit system of claim 20, wherein said circuit domain manager further identifies said route based at least in part on application of a shortest path algorithm.

22. The virtual circuit system of claim 20, wherein said circuit domain manager configures each said intermediate node within said route by specifying packet flow characteristics.

23. The virtual circuit system of claim 22, wherein said packet flow characteristics include an ingress identifier, an egress identifier, a priority level for said packets flowing through said virtual circuit, and a policing rate, said ingress identifier having a ingress physical port number and a ingress identifier type and said egress identifier having an egress physical port number and an egress identifier type.

24. The virtual circuit system of claim 23, wherein said policing rate for said source node is set at a value greater than or equal to said bandwidth.

25. The virtual circuit system of claim 23, wherein each said intermediate node on said route is IP layer 4 enabled, wherein said ingress identifier type and said egress identifier type are set to IP layer 4, wherein each of said packets exiting a first intermediate node is forwarded to a next intermediate node based on a pre-existing routing table within said intermediate nodes, and wherein each of said packets travelling through each of said intermediate nodes does so at a high priority.

26. The virtual circuit system of claim 23, wherein each said intermediate node on said route is DSCP enabled, wherein each of said packets has a specified DSCP value, and wherein each of said intermediate nodes is set to route said packets with said specified DSCP value at a high priority according to a pre-existing routing table within said intermediate node.

27. The virtual circuit system of claim 23, wherein each said intermediate node on said route is MPLS enabled, wherein each of said packets has a specified MPLS value, wherein at least one segment of said route has a pre-defined label switched path, and wherein each of said intermediate nodes within said segment is set to route said packets with said specified MPLS value at a high priority according to said pre-defined label switched path.

28. The system of claim 19, wherein said packet flow between said source node and said destination node is defined by a source node IP address, a destination node IP address, a source node UDP port number, a destination node UDP port number, and said bandwidth.

29. A method for enabling packets to flow at a predetermined QoS through a virtual circuit within a packet network including a network topology having a source node, a destination node and plurality of intermediate nodes selectively interconnected via a plurality of links, the method comprising the steps of:

detecting signalling protocol data transmitted and received by said source node within said network topology when said source node seeks to communicate with said destination node;

deriving virtual circuit attribute data from said signalling protocol data;

requesting establishment of said virtual circuit from a virtual circuit manager based on said virtual circuit attribute data without human intervention, said virtual circuit attribute data including QoS data characterized by a bandwidth requirement and a packet transfer delay limit between said source node and said destination node; and

establishing said virtual circuit based on said QoS data.

30. The method of claim 29, wherein said step of detecting occurs when said source node seeks to communicate with said destination node through a server connected to said packet network.

31. The method of claim 30, further comprising the step of:

terminating said virtual circuit if said virtual circuit cannot be established or maintained between said source node and said destination node in accordance with said QoS data.

32. The method of claim 30, wherein said server is a proxy in communication with an application server, and wherein said step of detecting includes the step of detecting at least a portion of said signalling protocol data from within said proxy after said proxy receives a portion of said signalling protocol data from said application server.

33. The method of claim 29, wherein said step of detecting occurs when said source node seeks to communicate with said destination node through a multimedia manager connected to said packet network, and wherein said step of detecting includes the step of detecting at least a portion of said signalling protocol data from said multimedia manager.

34. The method of claim 29, wherein said source node and said destination node are H.323 enabled, wherein said signalling protocol data is based on H.323, wherein said step of detecting includes the step of intercepting H.323 requests from said source node and H.323 responses to said H.323 requests, and wherein said step of deriving includes the step of processing said H.323 requests and said H.323 responses to derive said virtual circuit attribute data.

35. The method of claim 34, wherein said steps of detecting, deriving and requesting occur from within either said source node or said destination node.

36. The method of claim 34, wherein said steps of detecting, deriving and requesting occur from within a H.323 application server in communication with either said source node or said destination node.

37. The method of claim 36, further comprising the step of:

terminating transfer of said H.323 requests and said H.323 responses if said virtual circuit cannot be established or maintained between said source node and said destination node in accordance with said QoS data.

38. The method of claim 29, wherein said source node and said destination node are SIP enabled, wherein said signalling protocol data is based on SIP, wherein said step of detecting includes the step of intercepting SIP requests from said source node and SIP responses to said SIP requests, and wherein said step of deriving includes the step of processing said SIP requests and said SIP responses to derive said virtual circuit attribute data.

39. The method of claim 38, wherein said steps of detecting, deriving and requesting occur from within either said source node or said destination node.

40. The method of claim 38, wherein said steps of detecting, deriving and requesting occur from within a SIP application server in communication with either said source node or said destination node.

41. The method of claim 40, further comprising the step of:

terminating transfer of said SIP requests and said SIP responses if said virtual circuit cannot be established or maintained between said source node and said destination node in accordance with said QoS data.

42. The method of claim 29, wherein said step of establishing said virtual circuit includes the steps of:

identifying a proposed route through said intermediate nodes from said source node to said destination node for said packet flow;

configuring each said intermediate node along said proposed route to enable said virtual circuit;

testing said proposed route to verify that said proposed route will satisfy said QoS data; and

establishing said virtual circuit corresponding to said proposed route if said proposed route passes said step of testing.

43. The method of claim 42, wherein said step of identifying a proposed route includes the step of determining said network topology by dynamically sensing said plurality of links between said intermediate nodes directly from said intermediate nodes.

44. The method of claim 42, wherein said step of identifying includes the step of determining said proposed route based on a shortest path through said packet network between said source node and said destination node.

45. The method of claim 42, wherein said step of configuring includes the step of establishing packet flow characteristics for each said intermediate node, said packet flow characteristics including an ingress identifier, an egress identifier, a priority level for said packets flowing through said virtual circuit, and a policing rate, said ingress identifier having a ingress IP port number and a ingress identifier type and said egress identifier having an egress IP port number and an egress identifier type.

46. The method of claim 42, wherein said step of identifying a proposed route includes the step of testing each of said links along said proposed route to determine if $circuitBandwidth + B_{CoIP\_ij} \leq f_{ij} \times B_{ij}$, wherein:

circuitBandwidth is said bandwidth requirement;

B_{CoIP_ij} is a state variable for each link (i, j) between an intermediate node i and an intermediate node j, said state variable being equal to an aggregate bandwidth already committed to virtual circuits on said link (i, j);

f_{ij} is a predetermined maximum fraction of bandwidth capacity for each said link (i, j) reserved for virtual circuit connections; and

B_{ij} is a bandwidth capacity for each said link (i, j).

47. The method of claim 42, wherein said step of testing includes the step of determining a probability $D_{circuitDelayQuantile}$ for said proposed route and verifying that $D_{circuitDelayQuantile} \leq$ *circuitMaxDelay* for said proposed route, wherein:

circuitMaxDelay is said packet transfer delay limit;

$$D_{circuitDelayQuantile} = w_{circuitDelayQuantile} \times \left(\frac{MTU\_Size}{B_{min}} \right);$$

$$w_{circuitDelayQuantile} = \mu_{circuitHopCount} + \beta_{circuitHopCount}(circuitDelayQuantile) \times \sigma_{circuitHopCount};$$

MTU_Size is a maximum packet transfer unit size in bits;

B_{min} is a minimum bandwidth capacity B_{ij} for each said link (i, j) between an intermediate node i and an intermediate node j on said proposed route;

$$\mu_{circuitHopCount} = circuitHopCount \left(\frac{f_{max}}{2(1 - f_{max})} \right);$$

$$\sigma_{circuitHopCount} = \sqrt{circuitHopCount} \sqrt{\left(\frac{f_{max}}{2(1 - f_{max})} \right)^2 + \frac{f_{max}}{3(1 - f_{max})}};$$

circuitHopCount is a number of hops in said proposed route;

f_{max} is a maximum value for f_{ij} for every link on said proposed route, where f_{ij} is a predetermined maximum fraction of bandwidth capacity for each said link (i, j) reserved for virtual circuit connections; and

$\beta_{reservationHopCount}(reservationDelayQuantile)$ is a weighting factor obtained from a *circuitHopCount* and said *DcircuitDelayQuantile*, where said *circuitHopCount* is a number of said links in said proposed route and said *DcircuitDelayQuantile* is predetermined for said virtual circuit.

48. A system for enabling packets to flow at a predetermined QoS through a virtual circuit within a packet network, wherein said packet network includes a network topology having a

plurality of intermediate nodes selectively interconnected via a plurality of links, the system comprising:

a first circuit network manager operable to receive a virtual circuit request over said packet network, said virtual circuit request representing a request for a packet flow at said predetermined QoS between a source node in a first circuit domain and a destination node, said first circuit network manager being operable to identify any inter-circuit domain links between said first circuit domain and any other circuit domains within said network topology without being within said first circuit domain; and

a circuit domain manager in communication with said first circuit domain and operable to communicate with said circuit network manager to identify a route through said intermediate nodes from said source node to said destination node for said packet flow at said predetermined QoS, and to configure each said intermediate node along said route to enable said virtual circuit at said predetermined QoS without said circuit domain manager needing to be within said route, said predetermined QoS being characterized by a specified bandwidth, a maximum allowable delay for said packet flow between said source node and said destination node, and a probability that said packet flow between said source node and said destination node will exceed said maximum allowable delay during said virtual circuit's lifetime.

49. The system of claim 48, wherein said circuit domain manager identifies said route partially based on said network topology and whether each node in said route will deliver said predetermined QoS, and wherein said circuit domain manager is operable to determine said network topology by dynamically sensing said plurality of links between said nodes directly from said nodes.

50. The system of claim 49, wherein said circuit domain manager further identifies said route based at least in part on application of a shortest path algorithm.

51. The system of claim 48, wherein said destination node is located within a second circuit domain, wherein a second circuit domain manager is in communication with said second domain, and wherein said circuit network manager is further operable to identify an inter-circuit domain link between said first circuit domain and said second circuit domain, to generate a first proposed route request to said circuit domain manager to determine a first

domain route from said source node to said inter-circuit domain link, and to generate a second proposed route request to said second circuit domain manager domain to determine a second domain route from said inter-circuit domain link to said destination node, said route being formed through inter-connection of said first domain route and said second domain route.

52. The system of claim 51, wherein said circuit domain manager configures each said intermediate node within said route in said first circuit domain by specifying packet flow characteristics, and wherein said second circuit domain manager configures each said intermediate node within said route in said second circuit domain by specifying said packet flow characteristics.

53. The system of claim 52, wherein said packet flow characteristics include an ingress identifier, an egress identifier, a priority level for said packets flowing through said virtual circuit, and a policing rate, said ingress identifier having a ingress physical port number and a ingress identifier type and said egress identifier having an egress physical port number and an egress identifier type.

54. The system of claim 53, wherein said policing rate for said source node is set at a value greater than or equal to said specified bandwidth.

55. The system of claim 53, wherein each said intermediate node on said route is IP layer 4 enabled, wherein said ingress identifier type and said egress identifier type are set to IP layer 4, wherein each of said packets exiting a first intermediate node is forwarded to a next intermediate node on said route based on a pre-existing routing table within said first intermediate node, and wherein each of said packets travelling through each of said intermediate nodes does so at a high priority.

56. The system of claim 53, wherein each said intermediate node on said route is DSCP enabled, wherein each of said packets has a specified DSCP value, and wherein each of said intermediate nodes is set to route said packets with said specified DSCP value at a high priority according to a pre-existing routing table within said node.

57. The system of claim 53, wherein each said intermediate node on said route is MPLS enabled, wherein each of said packets has a specified MPLS value, wherein at least one

segment of said route has a pre-defined label switched path, and wherein each of said intermediate nodes within said segment is set to route said packets with said specified MPLS value at a high priority according to said pre-defined label switched path.

58. The system of claim 48, wherein said system further comprises a circuit client operable to generate said virtual circuit request and communicate said virtual circuit request to said circuit network manager, wherein said circuit network manager identifies to said circuit client when said virtual circuit can be set up, and wherein said circuit client is operable to permit said packet flow through said virtual circuit only when said route is capable of delivering said predetermined QoS.

59. The system of claim 58, wherein circuit client, said circuit network manager and said circuit domain manager reside within either of one of said intermediate nodes, said source node, or said destination node.

60. The system of claim 58, wherein said circuit client is implemented utilizing a data packet protocol that enables said circuit domain manager to identify said route and to enable said virtual circuit at least partially based on availability of said specified bandwidth.

61. The system of claim 48, wherein said packet flow at said predetermined QoS between said source node and said destination node is defined by a source node IP address, a destination node IP address, a source node UDP port number, a destination node UDP port number, and said specified bandwidth.

62. A method for enabling packets to flow at a predetermined QoS through a virtual circuit within a packet network including a network topology having a plurality of intermediate nodes selectively interconnected via a plurality of links, the method comprising the steps of:

receiving a request for establishment of said virtual circuit, said virtual circuit representing a request for said packet flow between a source node in a first circuit domain within said network topology and a destination node within said network topology;

identifying a proposed route through said intermediate nodes from said source node to said destination node through said packet network at said predetermined QoS, said QoS

characterized by a bandwidth, a maximum allowable delay for said packet flow between said source node and said destination node, and a probability that said packet flow between said source node and said destination node will exceed said maximum allowable delay during said virtual circuit's lifetime;

identifying inter-circuit domain links between said first circuit domain and any other circuit domains along said proposed route;

configuring each intermediate node on said proposed route to enable said virtual circuit at said predetermined QoS;

testing said proposed route to verify that said maximum allowable delay will not be exceeded; and

enabling the establishment of said virtual circuit corresponding to said proposed route if said proposed route passes said step of testing.

63. The method of claim 62, wherein said step of identifying said proposed route includes the step of determining a network topology by dynamically sensing said plurality of links between said intermediate nodes directly from said intermediate nodes.

64. The method of claim 63, wherein said step of identifying said proposed route includes the step of determining said proposed route based on said network topology and a shortest path through said packet network between said source node and said destination node.

65. The method of claim 62, wherein said step of configuring includes the step of establishing packet flow characteristics for each said intermediate node, said packet flow characteristics including an ingress identifier, an egress identifier, a priority level for said packets flowing through said virtual circuit, and a policing rate, said ingress identifier having a ingress IP port number and a ingress identifier type and said egress identifier having an egress IP port number and an egress identifier type.

66. The method of claim 62, wherein said step of identifying said proposed route includes the step of testing each of said links along said proposed route to determine if $circuitBandwidth + B_{CoIP\_ij} \leq f_{ij} \times B_{ij}$, wherein:

circuitBandwidth is said bandwidth;

B_{CoIP_ij} is a state variable for each link (i, j) between an intermediate node i and an intermediate node j, said state variable being equal to an aggregate bandwidth already committed to virtual circuits on said link (i, j);

f_{ij} is a predetermined maximum fraction of bandwidth capacity for each said link (i, j) reserved for virtual circuit connections; and

B_{ij} is a bandwidth capacity for each said link (i, j).

67. The method of claim 62, wherein said step of testing includes the step of determining a probability $D_{circuitDelayQuantile}$ for said proposed route and verifying that $D_{circuitDelayQuantile} \leq circuitMaxDelay$ for said proposed route, wherein:

circuitMaxDelay is said maximum allowable delay;

$$D_{circuitDelayQuantile} = w_{circuitDelayQuantile} \times \left(\frac{MTU\_Size}{B_{\min}}\right);$$

$$w_{circuitDelayQuantile} = \mu_{circuitHopCount} + \beta_{circuitHopCount}(circuitDelayQuantile) \times \sigma_{circuitHopCount};$$

MTU_Size is a maximum packet transfer unit size in bits;

B_{min} is a minimum bandwidth capacity B_{ij} for each said link (i, j) between an intermediate node i and an intermediate node j on said proposed route;

$$\mu_{circuitHopCount} = circuitHopCount\left(\frac{f_{\max}}{2(1-f_{\max})}\right);$$

$$\sigma_{circuitHopCount} = \sqrt{circuitHopCount}\sqrt{\left(\frac{f_{\max}}{2(1-f_{\max})}\right)^2 + \frac{f_{\max}}{3(1-f_{\max})}};$$

circuitHopCount is a number of hops in said proposed route;

f_{max} is a maximum value for f_{ij} for every link of said proposed route, where f_{ij} is a predetermined maximum fraction of bandwidth capacity for each said link (i, j) reserved for virtual circuit connections; and

$\beta_{reservationHopCount}(reservationDelayQuantile)$ is a weighting factor obtained from a *circuitHopCount* and said *DcircuitDelayQuantile*, where said *circuitHopCount* is a number of said links in said proposed route and said *DcircuitDelayQuantile* is predetermined for said virtual circuit.

68. A method of identifying a route satisfying a request for establishment of a virtual circuit between a source node in a first circuit domain and a destination node in a second circuit domain at a predetermined QoS, said source node and said destination node located in a packet network, wherein said packet network includes a plurality of intermediate nodes and a plurality of links between said intermediate nodes, the method comprising the steps of:

identifying an inter-circuit domain link between said first circuit domain and said second circuit domain;

determining a first proposed route within said first circuit domain from said source node to said inter-circuit domain link through said intermediate nodes;

determining a second proposed route within said second circuit domain from said inter-circuit domain link to said destination node through said intermediate nodes;

determining whether said first proposed route and said second proposed route will deliver a bandwidth requirement;

determining a maximum allowable delay for a packet flow between said source node and said destination node over said first proposed route and said second proposed route;

determining a probability that said packet flow will exceed said maximum allowable delay during said virtual circuit's lifetime; and

inter-connecting said first proposed route and said second proposed route to form said route if said route delivers said bandwidth requirement and does not exceed said maximum allowable delay for said packet flow.